Scale search powered apps with Elasticsearch, k8s and go

Maxime Boisvert

shopify

2016

Was a major year for me

Joined Shopify

Had a son

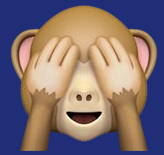
2017

Joined the search infrastructure team

Shopify + apps trying from various cloud provider to Kubernetes

Shopify is using an older version of Elasticsearch

Elasticsearch operations are manual



Good practices are not always in place

Growing number of Elasticsearch clusters

Search operator

Based on Elasticsearch, Kubernetes, Golang and Docker

Elasticsearch basics

- Cluster
- Node
- Index
- Replicas

Kubernetes basics

- Pod
- Deployment
- Statefulset
- Service
- Configmap
- Custom Resource Definition

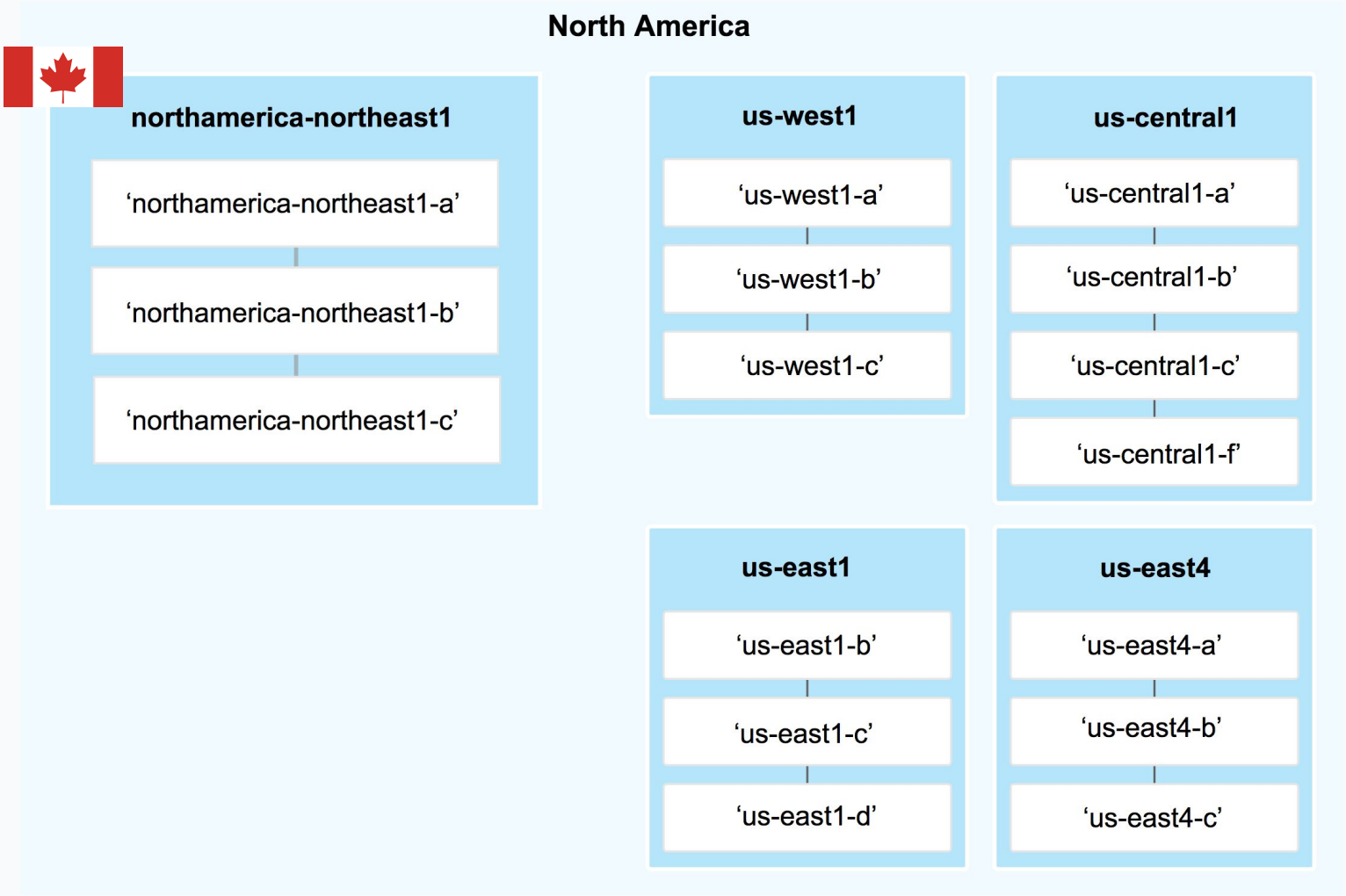
Mixing it together

Elasticsearch Kubernetes components

- Statefulset
  - Elasticsearch nodes
- Deployment
  - Snapshot pod with crontab and cli
- Configmap
  - Shared Elasticsearch and logging configuration
- Service
  - Endpoint: elasticsearch:9200
  - Discovery
  - Data

Multi-zone data resiliency

- Regions and Zones
- One Statefulset per zone
  - Persistent volume claim
- Allocation awareness setting


North America
northamerica-northeast1
‘northamerica-northeast1-a’
‘northamerica-northeast1-b’
‘northamerica-northeast1-c’
us-west1
‘us-west1-a’
‘us-west1-b’
‘us-west1-c’
us-central1
‘us-central1-a’
‘us-central1-b’
‘us-central1-c’
‘us-central1-f’
us-east1
‘us-east1-b’
‘us-east1-c’
‘us-east1-d’
us-east4
‘us-east4-a’
‘us-east4-b’
‘us-east4-c’

Implementation

Simple configuration: elasticsearch.yml

```
apiVersion: stable.shopify.io/v1
kind: Elasticsearch
metadata:
 name: app
spec:
 elastic-search-version: 6
 zones:
   - us-east1-b
   - us-east1-c
   - us-east1-d
 node-specs:
   - replicas: 3
     cpu-limit: "1"
     mem-limit: 2G
     data-volume-size: 10Gi
 snapshot:
   bucket-name: shopify-app-prod-es-snapshots
```

```
$ kubectl get all
NAME                                              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deploy/app--elasticsearch-snapshot-cronjob   1         1         1            1           3m

NAME                                                    DESIRED   CURRENT   READY     AGE
rs/app--elasticsearch-snapshot-cronjob-7d699dd658   1         1         1         3m

NAME                                                 DESIRED   CURRENT   AGE
statefulsets/app--elasticsearch-data-us-east1-b   1         1         3m
statefulsets/app--elasticsearch-data-us-east1-c   1         1         3m
statefulsets/app--elasticsearch-data-us-east1-d   1         1         3m

NAME                                                        READY     STATUS    RESTARTS   AGE
po/app--elasticsearch-data-us-east1-b-0                  1/1       Running   0          3m
po/app--elasticsearch-data-us-east1-c-0                  1/1       Running   0          3m
po/app--elasticsearch-data-us-east1-d-0                  1/1       Running   0          3m
po/app--elasticsearch-snapshot-cronjob-7d699dd658-f7p8w   1/1       Running   0          3m

NAME                                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
svc/app--elasticsearch             ClusterIP   10.187.243.214   <none>        9200/TCP   3m
svc/app--elasticsearch-data        ClusterIP   10.187.255.181   <none>        9300/TCP   3m
svc/app--elasticsearch-discovery   ClusterIP   10.187.242.125   <none>        9300/TCP   3m
```

Good practices are ~~not~~ always in place

Automate operations

Rolling update

- When Statefulset or Configmap change:
  - For each node from the previous configuration:
  - Wait for ready/green cluster
  - Stop allocation
  - Restart pod
  - Start allocation
- Downtime on master restart, no native way around
  - Elastic notification of maintenance email > This work will have minimum impact on your usage of the cluster. This change will involve a grow and shrink action on the cluster which you may notice and it may cause a reelection of the master node.

Scaling

- When configured replicas not the same as current of nodes:
  - While replicas != n of nodes
  - Wait for the cluster to be ready and green
  - Increment or decrement one by one
- Minimum master nodes
  - Split brain on scale up
  - Master lost on scale down

Elasticsearch operations are ~~manual~~ automated

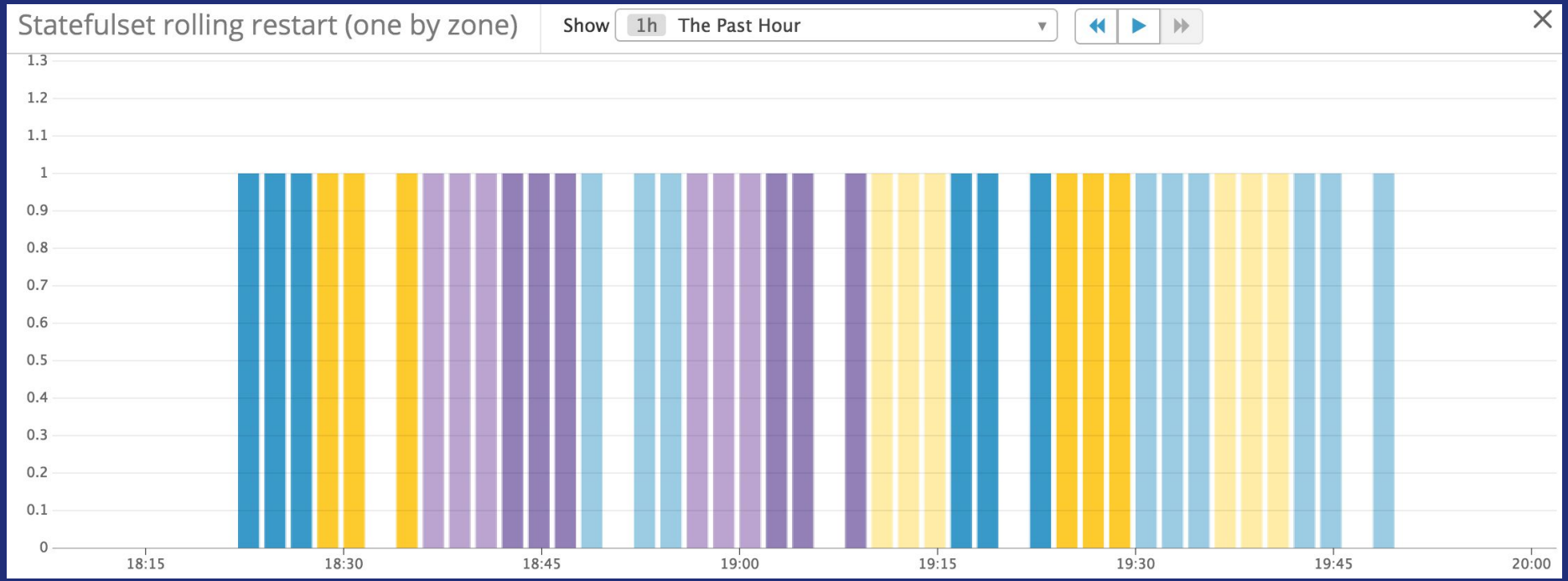
Snapshots for everyone

Snapshots

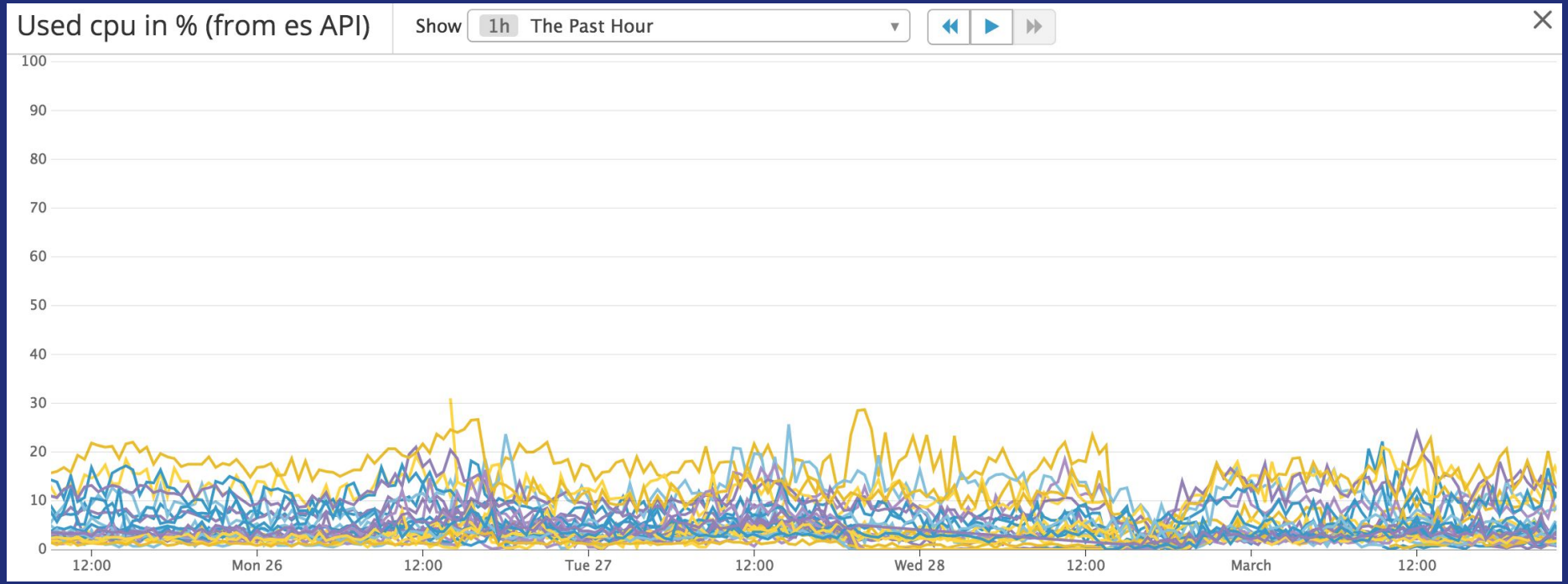
- Bucket
- Crontab
- CLI
- Purge

The big picture

Search operator: Datadog and Bugsnag



Elasticsearch: Datadog, Splunk and Kibana



2018

One year later

- Managed Elasticsearch clusters: tens
- Kubernetes pods: hundreds

- Shopify moving to Elasticsearch 6
- Clusters are up to date
- Many automated operations
- Good practices in place

Demo time

- Create
- Update
- Scale
- Delete

2017
Scaled search infra with k8s

2018
Search infra as a product (internally)

Thanks!

https://github.com/maxboisvert/confoo-montreal-2018